## AMENDMENT TO CLAIMS

Please amend claims 1, 3, 4, 5, 10, 20-26, 28-32, 34-35, 39, 42-46, 48-49, 57-58, 60, all as shown below. All pending claims are reproduced below, including those that remain unchanged.

1. (Currently Amended) A system to provide conversation states, comprising:

a first computing device capable of:

running a process on the first computing device; and

accepting a message during a conversation between the process running on the first computing device and another process, wherein the message is associated with a conversation state that operates to be invoked over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA);

a second computing device capable of:

maintaining [[a]] the conversation state requested by the message; and

storing information of the conversation state in memory on the second computing device; and

a conversation manager capable of:

identifying the location of the second computing device which maintains the conversation state requested by the message; and

providing the location and/or the information of the conversation state to the first computing device.

2. (Previously presented) The system according to claim 1, wherein:

the first and second computing devices form a cluster.

3. (Currently Amended) The system according to claim 1, wherein:

the conversation manager is capable of maintaining the locations of all conversation states in the system.

4. (Currently Amended) The system according to claim 1, wherein:

the information may include, a map of every conversation state leased, owned, or stored on the second computing device.

5. (Currently Amended) The system according to claim 1, wherein:

the first and second computing devices ~~can be~~ are the same computing device.

6. (Previously presented) The system according to claim 1, wherein:

the second computing device is capable of maintaining the information both in-memory and on persistent storage.

7. (Previously presented) The system according to claim 1, wherein:

the conversation manager is capable of designating the second computing device as the primary and replicating the information on the second computing device to a third computing device.

8. (Previously presented) The system according to claim 7, wherein:

the conversation manager is capable of routing to the third computing device and setting it as the new primary when the second computing device fails.

9. (Previously presented) The system according to claim 1, wherein:

the conversation manager is capable of periodically determining the availability of the second and third computing devices.

10. (Currently Amended) A system to provide conversation for Web service, comprising:

a conversation partner, which is a process, capable of providing a message during a conversation between the conversation partner and a process running on a first computing device;

said first computing device capable of accepting a message during the conversation with the conversation partner, wherein the message is associated with a conversation state that operates to be invoked over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA);

a second computing device capable of:

maintaining [[a]] the conversation state requested by the message; and

storing information of the conversation state in memory on the second computing device; and

a conversation manager capable of:

identifying the location of the second computing device which maintains the conversation state requested by the message; and

providing the location and/or the information of the conversation state to the first computing device.

11. (Original) The system according to claim 10, wherein:

the message includes a conversation ID.

12-19. (Canceled).

20. (Currently Amended) The system according to claim 11, wherein:

the first computing device is capable of contacting the conversation manager to determine the location of the conversation state requested by the message using the conversation ID.

21. (Currently Amended) The system according to claim 10, wherein:

the first computing device is capable of answering a request for the conversation state directly without contacting the conversation manager if it owns such state.

22. (Currently Amended) The system according to claim 10, wherein:

the conversation manager is capable of accepting a request for the location of the conversation state from the first computing device.

23. (Currently Amended) The system according to claim 11, wherein:

the conversation manager is capable of providing the location and/or the information of the conversation state to the first computing device requesting it based on the conversation ID.

24. (Currently Amended) The system according to claim 10, wherein:

the first computing device is capable of accepting the location of the conversation state from the conversation manager.

25. (Currently Amended) The system according to claim 10, wherein:

the first computing device is capable of invoking the conversation state on the second computing device in order to respond to the conversation message received.

26. (Currently Amended) The system according to claim 10, wherein:

the conversation manager is capable of sharing the conversation state with at least two conversations.

27. (Previously presented) The system according to claim 10, wherein:

the conversation manager is capable of tracking a participating Web service that initiates the conversation.

28. (Currently Amended) The system according to claim 27, wherein:

the conversation manager is capable of sharing the conversation state with at least two Web services and joining the sessions of these services.

29. (Currently Amended) A method to provide a conversation for a Web service, comprising:

maintaining a conversation state on a computing device;

storing information of the conversation state in memory on the computing device;

accepting a message requesting the conversation state during a conversation between two processes;

contacting a conversation manager to determine the location of the conversation state requested by the message;

accepting the location and/or the information of the conversation state from the conversation manager; and

invoking the conversation state over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message.

30. (Currently Amended) A method to provide a conversation for a Web service, comprising:

maintaining a conversation state on a computing device;

storing information of the conversation state in memory on the computing device;

accepting a message requesting the conversation state during a conversation between two processes; and

invoking the conversation state over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message received directly at the computing device without contacting a conversation manager.

31. (Currently Amended) The method according to claim 29, further comprising:

maintaining the locations of all conversation states in the system on the conversation manager.

32. (Currently Amended) The method according to claim 29, further comprising:

maintaining on [[a]] the computing device its conversation state information, which may include, a map of every state leased, owned, or stored on it.

33. (Canceled).

34. (Currently Amended) The method according to claim 32, further comprising:

maintaining the conversation state information on the computing device both in-memory and on persistent storage.

35. (Currently Amended) The method according to claim 32, further comprising:

designating the computing device as the primary and replicating the conversation state information on the computing device to another computing device.

36. (Previously presented) The method according to claim 35, further comprising:

routing to the another computing device; and

setting it as the new primary when the current primary computing device fails.

37. (Previously presented) The method according to claim 29, further comprising:

determining the availability of the computing device~~s~~ periodically.

38. (Canceled).

39. (Currently Amended) The method according to claim 29, further comprising:

accepting request for the location of the conversation state from a computing device; and providing the location of the conversation state to the computing device requesting it.

40. (Previously presented) The method according to claim 29, further comprising:

sharing the state with at least two conversations.

41. (Previously presented) The method according to claim 29, further comprising:

tracking a participating Web service that initiates the conversation.

42. (Currently Amended) The method according to claim 41, further comprising:

sharing the conversation state with at least two Web services; and

joining the sessions of these services.

43. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

maintain a conversation state on a computing device;

store the information of the conversation state in memory on the computing device;

accept a message requesting the state during a conversation between two processes;

contact a conversation manager to determine the location of the conversation state requested by the message;

accept the location and/or the information of the conversation state from the conversation manager; and

invoke the conversation state over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message.

44. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

maintain a conversation state on a computing device;

store information of the conversation state in memory on the computing device;

accept a message requesting the conversation state during a conversation between two processes; and

invoke the conversation state over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message received directly at the computing device without contacting a conversation manager.

45. (Currently Amended) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

maintain the locations of all conversation states in the system on the conversation manager.

46. (Currently Amended) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

maintain on the computing device information, which may include, a map of every conversation state leased, owned, or stored on it.

47. (Canceled).

48. (Currently Amended) The machine readable medium of claim 46, further comprising instructions that when executed cause the system to:

maintain the conversation state information on the computing device both in-memory and on persistent storage.

49. (Currently Amended) The machine readable medium of claim 48, further comprising instructions that when executed cause the system to:

designating the computing device as the primary and replicating the conversation state information on the computing device to another computing device.

50. (Previously presented) The machine readable medium of claim 49, further comprising instructions that when executed cause the system to:

route to the another computing device; and

set it as the new primary when the current primary computing device fails.

51. (Previously presented) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

check for the availability of the computing devices periodically.

52. (Canceled).

53. (Previously presented) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

accept request for the location of the state from a computing device; and

provide the location of the state to the computing device requesting it.

54. (Previously presented) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

share the state with at least two conversations.

55. (Previously presented) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

track a participating Web service that initiates the conversation.

56. (Currently Amended) The machine readable medium of claim 55, further comprising instructions that when executed cause the system to:

share the conversation state with at least two Web services; and

join the sessions of these services.

57. (Currently Amended) A system for handling conversation, comprising:

means for maintaining a conversation state on a computing device;

means for storing information of the conversation state in memory on the computing device;

means for accepting a message requesting the conversation state during a conversation between two processes;

means for contacting a conversation manager to determine the location of the conversation state requested by the message;

means for accepting the location and/or the information of the conversation state from the conversation manager; and

means for invoking the conversation state over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message.

58. (Currently Amended) A computer data signal embodied in a transmission medium, comprising:

a code segment including instructions to maintain a conversation state on a computing device;

a code segment including instructions to store information of the conversation state in memory on the computing device;

a code segment including instructions to accept a message requesting the conversation state during a conversation between two processes;

a code segment including instructions to contact a conversation manager to determine the location of the conversation state requested by the message;

a code segment including instructions to accept the location and/or the information of the conversation state from the conversation manager; and

a code segment including instructions to invoke the conversation state over an application service such as a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message.

59. (Previously presented) The system according to claim 1, wherein:

the conversation can be within the context of a business application.

60. (Currently Amended) The system according to claim 1, wherein:

the conversation state can be one of: a program, an application, a service, and a database instance.